An Adaptive Projection Method for the Incompressible Euler Equations

Ann S. Almgren *
John B. Bell *
Lawrence Livermore National Laboratory
Livermore, CA 94550

Phillip Colella †
University of California at Berkeley
Berkeley, CA 94720

Louis H. Howell *
Lawrence Livermore National Laboratory
Livermore, CA 94550

Abstract

In this paper we present a method for solving the time-dependent incompressible Euler equations on an adaptive grid. The method is based on a projection formulation in which we first solve convection equations to predict intermediate velocities, and then project these velocities onto a space of approximately divergence-free vector fields. Our treatment of the convection step uses a specialized second-order upwind method for differencing the nonlinear convection terms that provides a robust treatment of these terms suitable for inviscid flow.

Our approach to adaptive refinement uses a nested hierarchy of grids with simultaneous refinement of the grids in both space and time. The integration algorithm on the grid hierarchy is a recursive procedure in which a coarse grid is advanced, fine grids are advanced multiple steps to reach the same time as the coarse grid and the grids are then synchronized. We will describe the integration algorithm in detail, with emphasis on the projection used to enforce the incompressibility constraint. Numerical examples are presented to demonstrate the convergence properties of the method and to illustrate the behavior of the method at the interface between coarse and fine grids. An additional example demonstrates the performance of the method on a more realistic problem.

*This work of these authors was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract W-7405-Eng-48. Support under contract W-7405-Eng-48 was provided by the AMS and HPCC Programs of the DOE Office of Scientific Computing and by the Defense Nuclear Agency under IACRO 93-817.

†Research supported at UC Berkeley by DARPA and the National Science Foundation under grant DMS-8919074; by a National Science Foundation Presidential Young Investigator award under grant ACS-8958522; and by the Department of Energy High Performance Computing and Communications Program under grant DE-FGO3-92ER25140.

Introduction

In this paper we develop a local adaptive mesh refinement algorithm for variable-density, inviscid, incompressible flow based on a second-order projection method. The equations governing this flow are:

$$U_t + (U \cdot \nabla)U = -\frac{1}{\rho}\nabla p + F, \tag{1.1}$$

$$\rho_t + (U \cdot \nabla)\rho = 0, \tag{1.2}$$

$$\nabla \cdot U = 0 \tag{1.3}$$

where U, ρ, and p represent the velocity, density, and pressure, respectively, and F represents any external forces. We denote the x and y components of velocity by u and v, respectively. The development of

the single grid second-order projection methodology for the incompressible Navier Stokes equations is discussed in a series of papers by Bell, Colella and Glaz [4], Bell, Colella and Howell [5], and Almgren, Bell and Szymczak [1]. The method discussed here is an adaptive version of the algorithm presented by Almgren et al [1], generalized to include finite amplitude density variation as discussed in Bell and Marcus [6]. The basic methodology uses a second-order upwind method for the treatment of the nonlinear convective terms in (1.1)-(1.2). The algorithms presented in those papers were motivated by a desire to apply higher-order upwind methods developed for gas dynamics to incompressible flow. In particular, they use a specialized version of the unsplit second-order upwind methodology introduced for gas dynamics by Colella [12]. The upwind methodology provides a robust discretization of the convective terms that avoids any stability restriction for inviscid flow.

The focus of this paper is on incorporating a local adaptive mesh refinement algorithm (AMR) into the basic projection methodology. This algorithm uses a hierarchical grid approach first developed by Berger and Oliger [9] for hyperbolic partial differential equations. This approach has been demonstrated to be highly successful for high speed flow by Berger and Colella [8] in two dimensions and by Bell et al [3] in three dimensions. AMR is based on a sequence of nested grids with successively finer spacing in both time and space. These fine grids are recursively embedded in coarser grids until the solution is sufficiently resolved. An error estimation procedure automatically gauges the accuracy of the solution and grid generation procedures dynamically create or remove rectangular fine grid patches as resolution requirements change.

Before describing the adaptive algorithm we will review, in the next section, the basic fractional step scheme for a single grid. In the third section we describe the recursive time stepping procedure for the adaptive algorithm. Other aspects of the adaptive algorithm are also sketched. The fourth section describes in more detail the approximation of the projection and a multigrid algorithm for solving the associated linear system of equations. In the fifth section we present computational results obtained with the method.

Single Grid Projection Algorithm

In this section we review the basic fractional step scheme for the case of a single uniform grid. The reader is referred to [1], [4], and [7], for a more detailed description. The algorithm uses a staggered grid scheme in which velocity and density are given at cell centers and are denoted by U_{ij}^n and ρ_{ij}^n respectively. Pressure is specified at cell corners and is staggered in time; thus, pressure is denoted by $p_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}}$.

The single grid algorithm for solving the system (1.1)-(1.3) is a fractional step scheme having two parts. First, we solve the advection equations (1.1)-(1.2) for the updated density and an intermediate velocity field without strictly enforcing the incompressibility constraint. Then, we project this intermediate field onto the space of (approximately) discretely divergence-free vector fields.

For the advection step we solve

$$\frac{\rho^{n+1}-\rho^n}{\Delta t}+[(U\cdot\nabla)\rho]^{n+\frac{1}{2}}=0 \qquad (2.1)$$

and

$$\frac{U^*-U^n}{\Delta t}+[(U\cdot\nabla)U]^{n+\frac{1}{2}}=-\frac{1}{\rho^{n+\frac{1}{2}}}\nabla p^{n-\frac{1}{2}}+F \qquad (2.2)$$

for the intermediate velocity U^* and the updated density ρ^{n+1}. The method uses an unsplit second-order upwind predictor-corrector scheme for evaluating the advective derivatives in (2.1)-(2.2). For this step the pressure gradient is evaluated at $t^{n-\frac{1}{2}}$ and is treated as a source term in (2.2), with $\rho^{n+\frac{1}{2}}=\frac{1}{2}(\rho^n+\rho^{n+1})$.

In the predictor we extrapolate the velocity and density to the cell edges at $t^{n+\frac{1}{2}}$ using a second-order Taylor series expansion. We denote by W the vector (U,ρ) and define S_{ij} to be the source vector $(F_{ij}-\frac{1}{\rho^{n+\frac{1}{2}}}(Gp)_{ij}^{n-\frac{1}{2}},0)$ where Gp is a second-order approximation to ∇p at cell centers, computed from the corner values of p. With this notation, the predictor step for edge $(i+\frac{1}{2},j)$ gives

$$W_{i+\frac{1}{2},j}^{n+\frac{1}{2},L}=W_{ij}^n+\frac{\Delta x}{2}W_{x,ij}^n+\frac{\Delta t}{2}W_{t,ij}^n$$

extrapolating from (i,j), and

$$W_{i+\frac{1}{2},j}^{n+\frac{1}{2},R}=W_{i+1,j}^n-\frac{\Delta x}{2}W_{x,i+1,j}^n+\frac{\Delta t}{2}W_{t,i+1,j}^n$$

extrapolating from $(i+1,j)$, with analogous formulae for the other edges. The differential equations (1.1) and (1.2) are then used to eliminate the time derivatives to obtain

$$W^{n+\frac{1}{2},L}_{i+\frac{1}{2},j} = W^n_{ij} + \left(\frac{\Delta x}{2} - \frac{u_{ij}\Delta t}{2}\right) W^n_{x,ij} \qquad (2.3)$$
$$-\frac{\Delta t}{2}(\widehat{vW_y})_{ij} - \frac{\Delta t}{2} S_{ij}$$

$$W^{n+\frac{1}{2},R}_{i+\frac{1}{2},j} = W^n_{i+1,j} - \left(\frac{\Delta x}{2} + \frac{u_{i+1,j}\Delta t}{2}\right) W^n_{x,i+1,j}$$
$$-\frac{\Delta t}{2}(\widehat{vW_y})_{i+1,j} - \frac{\Delta t}{2} S_{i+1,j} \qquad (2.4)$$

In evaluating these terms the first-order derivatives normal to the edge (in this case W_x) are evaluated using a monotonicity-limited fourth-order centered-difference slope approximation [11]. The limiting is done on the components of W individually.

The transverse derivative terms ($\widehat{vW_y}$ in this case) are evaluated using an upwind difference. In particular, we define

$$\widehat{W}^B_{i,j+\frac{1}{2}} = W^n_{ij} + \left(\frac{\Delta y}{2} - \frac{v_{ij}\Delta t}{2}\right) W_{y,ij}$$

$$\widehat{W}^T_{i,j+\frac{1}{2}} = W^n_{i,j+1} - \left(\frac{\Delta y}{2} + \frac{v_{i,j+1}\Delta t}{2}\right) W_{y,i,j+1}$$

where W_y are limited slopes in the y direction, with similar formulae for edge $(i, j-\frac{1}{2})$. A procedure analogous to the corrector described below is applied to the $\widehat{W}$ states to evaluate a convective difference approximation to $(\widehat{vW_y})$.

In the corrector step we first resolve the ambiguity in the edge values. The convective part of (1.1) corresponding to the velocity normal to the edge is of the form $u_t + uu_x = \textit{source terms}$. This suggests the following upwind determination of the normal velocity component:

$$u_{i+\frac{1}{2},j} = \begin{cases} u^L & \text{if } u^L > 0, u^L + u^R > 0 \\ 0 & \text{if } u^L \le 0, u^R \ge 0 \\ u^R & \text{otherwise} \end{cases}$$

(We suppress the $i+\frac{1}{2}, j$ spatial indices on left and right states here and in the next equation.) We now upwind W based on $u_{i+\frac{1}{2},j}$:

$$W_{i+\frac{1}{2},j} = \begin{cases} W^L & \text{if } u_{i+\frac{1}{2},j} > 0 \\ \frac{1}{2}(W^L + W^R) & \text{if } u_{i+\frac{1}{2},j} = 0 \\ W^R & \text{if } u_{i+\frac{1}{2},j} < 0 \end{cases}$$

Finally, we use these upwind values to form an approximation to the convective derivatives in (1.1)-(1.2)

$$uW_x + vW_y \approx$$
$$\frac{1}{2}(u_{i+\frac{1}{2},j} + u_{i-\frac{1}{2},j})(W_{i+\frac{1}{2},j} - W_{i-\frac{1}{2},j}) +$$
$$\frac{1}{2}(v_{i,j+\frac{1}{2}} + v_{i,j-\frac{1}{2}})(W_{i,j+\frac{1}{2}} - W_{i,j-\frac{1}{2}})$$

Using equations (2.1)-(2.2) we now compute the new approximations to ρ and U^*. The velocity field U^* computed in the first step is not, in general, divergence-free. The projection step approximately enforces the incompressibility constraint. A vector field decomposition is applied to $\frac{U^*-U^n}{\Delta t}$ to obtain the new velocity field and an update for the pressure. In particular, if $\mathbf{P}$ represents the projection then

$$\frac{U^{n+1} - U^n}{\Delta t} = \mathbf{P}\left(\frac{U^* - U^n}{\Delta t}\right) \qquad (2.5)$$

$$\frac{1}{\rho^{n+\frac{1}{2}}}\nabla p^{n+\frac{1}{2}} = \frac{1}{\rho^{n+\frac{1}{2}}}\nabla p^{n-\frac{1}{2}} + (\mathbf{I} - \mathbf{P})\left(\frac{U^* - U^n}{\Delta t}\right),$$

where $\rho^{n+\frac{1}{2}} = \frac{1}{2}(\rho^n + \rho^{n+1})$. (Note that the vector field we project is not U^*; it is an approximation to U_t.) Discretely, the projection is computed by solving for the gradient component of

$$\frac{U^* - U^n}{\Delta t}$$

which we denote by $\nabla\phi$. We determine ϕ by solving

$$\nabla \cdot \left(\frac{1}{\rho^{n+\frac{1}{2}}}\nabla\phi\right) = D\frac{U^* - U^n}{\Delta t} \qquad (2.6)$$

where D is a second-order discrete divergence operator that approximates divergence at cell corners from surrounding cell-centered velocities. This elliptic equation is discretized using a standard nine-point finite difference method analogous to the finite element method with bilinear basis elements. (The discretization is described in greater detail in a later section.) We then define

$$\frac{U^{n+1} - U^n}{\Delta t} = \frac{U^* - U^n}{\Delta t} - \frac{1}{\rho^{n+\frac{1}{2}}} G\phi$$

and

$$p^{n+\frac{1}{2}} = p^{n-\frac{1}{2}} + \phi$$

We note that this is not a discrete orthogonal projection; in fact, $DU^{n+1} \neq 0$. However, the incompressibility constraint is approximated to second-order accuracy and the overall algorithm is stable. The reader is referred to Almgren et al [1] for a detailed discussion of this approximation to the projection.

We also note that the upwind method is an explicit difference scheme and, as such, requires a time-step restriction. A linear, constant-coefficient analysis shows that for stability we must require

$$\max_{ij}\left(\frac{|u_{ij}|\Delta t}{\Delta x}, \frac{|v_{ij}|\Delta t}{\Delta y}\right) = \sigma \leq 1,$$

where σ is the CFL number. This time-step restriction for the Godunov method is used to set the time step for the overall algorithm.

Adaptive Mesh Refinement

In this section we describe the local adaptive mesh algorithm based on the single grid projection algorithm described in the previous section. The adaptive algorithm uses a hierarchy of nested rectangular grids on which the equations (1.1)-(1.3) are discretized. The grid hierarchy is defined logically by D_ℓ, $\ell = 0, ..., \ell_{max}$, where each D_ℓ is a collection of points in $\mathbf{Z}^2$ and $\ell_{max} + 1$ is the number of levels. Each D_ℓ is represented as the union of (potentially overlapping) rectangles in $\mathbf{Z}^2$; the calculations are organized around performing operations on each rectangular grid. However, the results obtained from the calculation are independent of the particular choice of decomposition. Each point in D_ℓ indexes a finite difference cell with mesh spacing $\Delta x^\ell = \Delta y^\ell = h^\ell$ We identify cells at successive levels ℓ, $\ell + 1$ by a map $\varphi : D_{\ell+1} \rightarrow D_\ell$, $\varphi(\mathbf{i}) = \mathbf{i}/r_\ell$, where r_ℓ is an even integer, called the *refinement ratio* (usually 2 or 4). Thus $D_{\ell+1}$ corresponds to a grid refined by a factor of r_ℓ in each coordinate direction, $h^{\ell+1} = \frac{h^\ell}{r_\ell}$ and

$$\varphi^{-1}(\mathbf{i}) = \{r_\ell \mathbf{i} + \mathbf{p} : \mathbf{p} = (0,0), ..., (r_\ell - 1, r_\ell - 1)\}$$

is the set of cells in $D_{\ell+1}$ contained in cell $\mathbf{i}$ in D_ℓ. The grids are nested, in the sense that $\varphi(D_{\ell+1}) \subset D_\ell$, and $\varphi^{-1}(\varphi(D_{\ell+1})) = D_{\ell+1}$. In addition, we require them to be *properly nested*, i.e., for $\ell > 0$, a one-cell wide buffer separates $\varphi(D_{\ell+1})$ from the boundary of D_ℓ, except for those parts of the boundary coinciding with boundaries of the physical domain of the problem. Refinement is done in time as well as in space: if Δt_ℓ is the time step at level ℓ, then $\Delta t_{\ell+1} = \Delta t_\ell / r_\ell$. In particular, the time steps are chosen so that the explicit time step restriction for the single grid algorithm is enforced throughout the grid hierarchy.

Integration Step

Integration of (1.1)-(1.3) on the adaptive hierarchy is a multistep recursive process. We will describe the algorithm in terms of the integration of grids at level ℓ. At time t^ℓ, we have $U^\ell(t^\ell)$, $\rho^\ell(t^\ell)$ on all of the cells in D_ℓ. In addition, we assume that $p^\ell(t^\ell - \frac{\Delta t^\ell}{2})$ is defined at all corners of cells in D_ℓ that are not located on interior boundaries between level ℓ and level $\ell - 1$ cells. In the following, we will refer to this collection of points as *level ℓ nodes*. At the end of the time step, we will have performed the updates $U^\ell(t^\ell) \rightarrow U^\ell(t^\ell + \Delta t^\ell)$, $\rho^\ell(t^\ell) \rightarrow \rho^\ell(t^\ell + \Delta t^\ell)$, $p^\ell(t^\ell - \frac{\Delta t^\ell}{2}) \rightarrow p^\ell(t^\ell + \frac{\Delta t^\ell}{2})$ A complete integration cycle on the entire grid hierarchy is given by invoking the algorithm for the coarsest level ($\ell = 0$) grid. We note that at the beginning of the time step we assume that the grids have been synchronized so that for each cell at level ℓ covered by level $\ell + 1$ grids the level ℓ value is the average of the values in the level $\ell + 1$ cells that cover it.

Step 1. For each grid at level ℓ we apply the upwind advection scheme described in the second section to compute $U^{\ell,*}$ and $\rho^\ell(t^\ell + \Delta t^\ell)$. When the advection calculation on each rectangular grid is performed, data are provided on the grid to be integrated as well as on a border of cells sufficiently wide to advance the solution. Data are copied from other level ℓ grids wherever such data are available; otherwise, data interpolated in space and time from coarser grids are used.

Step 2. We project $U^{\ell,*}$ onto its (approximately) divergence-free part to obtain a initial approximation to $U^\ell(t^\ell + \Delta t^\ell)$, by solving an equation analogous to (2.6). This is done using Dirichlet boundary conditions for ϕ in the elliptic solve interpolated from the level $\ell - 1$ grids. We refer to this projection as a *level projection*, because it is used to update the velocities and pressure on all the grids at a single level; the data

at every other level remains unchanged after a level projection. By itself, this projection is not sufficient to account properly for the coupling between levels in the elliptic equation defining the projection, inasmuch as it only forces matching of Dirichlet data at the coarse–fine interface, rather than both Dirichlet and Neumann data. This will be remedied by performing a second projection in Step 3 below.

We compute the scalar field ϕ^ℓ by solving the elliptic equation

$$L^\ell(\phi^\ell, \phi^{\ell-1}) = D^\ell(\frac{U^{*,\ell} - U^\ell(t^\ell)}{\Delta t^\ell})$$

where $L^\ell(\cdot)$ is the discretization of the second–order elliptic operator

$$\nabla \cdot \left(\frac{1}{\rho}\nabla(\cdot)\right) \tag{3.1}$$

on the level ℓ nodes, and D^ℓ is the associated divergence operator. The values for ρ in (3.1) are the average of $\rho^\ell(t^\ell)$ and $\rho^\ell(t^\ell + \Delta t^\ell)$. The boundary conditions for ϕ^ℓ on the boundary between the level ℓ and level $\ell - 1$ are given by interpolating the values of

$$\frac{p^{\ell-1}(t^{\ell-1} + \frac{1}{2}\Delta t^{\ell-1}) - p^{\ell-1}(t^{\ell-1} - \frac{1}{2}\Delta t^{\ell-1})}{r_{\ell-1}},$$

onto the fine grid boundary points using linear interpolation in space. Here $t^{\ell-1}$ is the last time for which we have data at level $\ell - 1$. Since ϕ is an estimate of the correction to the pressure, this amounts to using a piecewise constant interpolation in time of that correction from the coarse grid onto the fine grid. At boundary points corresponding to the physical boundary, the usual physical boundary conditions for ϕ are used. These are the only boundary conditions applied at level $\ell = 0$. The correction to the velocity field is then performed using the gradient of ϕ.

$$\frac{U^\ell(t^\ell + \Delta t^\ell) - U^\ell(t^\ell)}{\Delta t^\ell} = \frac{U^{\ell,*} - U^\ell(t^\ell)}{\Delta t^\ell} - \frac{G\phi^\ell}{\rho^\ell(t^\ell + \frac{1}{2}\Delta t^\ell)}$$

and

$$p^\ell(t^\ell + \frac{1}{2}\Delta t^\ell) = p^\ell(t^\ell - \frac{1}{2}\Delta t^\ell) + \phi^\ell$$

Step 3. If there are grids at level $\ell + 1$ then the integration step is called recursively to advance the level $\ell + 1$ grids. After the level $\ell + 1$ grids complete r_ℓ consecutive time steps, the velocity and density data for all levels greater than or equal to ℓ have been advanced to time $t^\ell + \Delta t^\ell$. However, the corrections to the pressure field $p^{\ell'}$, $\ell' > \ell$ were computed using boundary conditions that had been interpolated from the level ℓ solution, with no correction of the level ℓ solution due to the presence of the finer grids. This is analogous to solving Poisson's equation on a locally refined mesh by first solving on the coarse mesh, then interpolating Dirichlet boundary conditions for the fine mesh solution. In that case, it is well-known that the accuracy on both the coarse and fine meshes is limited to the accuracy on the coarse mesh alone, i.e., all benefit from refinement is lost. The error is caused by a jump in the normal derivative at the coarse–fine boundary that is on the order of the truncation error in the coarse grid. To eliminate this error, one must iterate between the coarse and fine grids. We perform the analogous procedure in the present setting by performing a projection step on the velocity field on all levels $\ell' \geq \ell$. We call this projection a *sync projection*, since it synchronizes the solution at all levels.

The sync projection is defined as follows. We solve the following system of equations for $\phi^{\ell'}$, $\forall \ell' \geq \ell$:

$$L^{\ell',\ell'+1}(\phi^{\ell'+1}, \phi^{\ell'}, \phi^{\ell'-1}) = D^{\ell',\ell'+1}(U^{\ell'}, U^{\ell'+1}). \tag{3.2}$$

Here, $D^{\ell',\ell'+1}(U^{\ell'}, U^{\ell'+1})$ is the discrete divergence operator defined on level ℓ' nodes. Away from the nodes that lie on the level $\ell' + 1$ boundary, $D^{\ell',\ell'+1}$ is just the usual single grid divergence operator. On the boundary nodes, $D^{\ell',\ell'+1}$ uses a special stencil that involves adjacent level $\ell'+1$ velocity values. Similarly, $L^{\ell',\ell'+1}$ is the discretization of the operator (3.1) on the level ℓ' nodes. It is linear in each of its arguments, and uses $\rho^{\ell'}(t^\ell + \Delta t^\ell)$. Away from the level $\ell' + 1, \ell'$ boundaries, it is the uniform grid discretization; near those boundaries, it uses values from the both levels to ensure continuity of ϕ and its normal derivative across level boundaries. The equations (3.2) form a linear system for the $\phi^{\ell'}$s; we solve this linear system using a hierarchical mesh multigrid algorithm similar to those developed by Almgren, Buttke and Colella [2] and by Howell [13]. We then compute corrected values for the velocities:

$$U^{\ell'}(t^\ell + \Delta t^\ell) := U^{\ell'}(t^\ell + \Delta t^\ell) - \frac{G^{\ell'}(\phi^{\ell'-1}, \phi^{\ell'})}{\rho^{\ell'}(t^\ell + \Delta t^\ell)}.$$

This step corrects the failure of the velocities on the

finer grids to satisfy the divergence–free condition at coarse–fine boundaries (most noticeably on the level $\ell, \ell+1$ boundary since the other levels have been matched within the recursive calls to the integration step at the finer levels). Finally, we correct the pressures:

$$p^{\ell'}(t^\ell + \Delta t^\ell - \frac{1}{2}\Delta t^{\ell'}) :=$$

$$p^{\ell'}(t^\ell + \Delta t^\ell - \frac{1}{2}\Delta t^{\ell'}) - \phi^{\ell'}$$

Note that for a three level calculation at the end of a level 0 time step we actually perform two sync projections. The first synchronizes levels 1 and 2. The second synchronizes levels 0, 1, and 2. The reason for this is that the projection step of the algorithm as defined in the previous section occurs at half time levels. In the adaptive algorithm the sync projection at level ℓ is a correction to the approximation to U_t at $t^\ell + \Delta t^\ell/2$. Thus, the two sync projections represent corrections at different times. Once we have completed the sync projection and updated the velocities and pressure, we average the solution from the level ℓ' grids recursively onto the coarser grids, down to level ℓ.

Creating the Grid Hierarchy

The initial creation of the grid hierarchy and the subsequent regridding operations in which the grids are dynamically changed to reflect changing flow conditions use the same procedures as were used by Bell et al [3] for the hyperbolic case. The grid hierarchy is constructed using an error estimation criterion to determine where additional resolution is required. (For the examples shown in the last section we have used simple error estimation methods based on the magnitude of the vorticity or the density gradient; however, more precise approaches such as using Richardson extrapolation to estimate error can be used.) Given grids at level ℓ we use the error estimation procedure to tag cells where the error is above a given tolerance. The tagged cells are grouped into rectangular patches using the clustering algorithm given in Berger et al [10]. These rectangular patches are refined to form the grids at the next level. The process is repeated until either an error tolerance criterion is satisfied or a specified maximum level is reached. The proper nesting requirement is imposed at this stage, namely that the union of level $\ell+1$ grids be properly contained within the union of level ℓ grids (except at the boundary of the physical domain where all levels can be refined up to the edge). This ensures that all coarse-fine interfaces are between successive levels; a level $\ell+2$ grid never directly interacts with a level ℓ grid.

As the computation proceeds a regridding algorithm is periodically used to adjust the placement of the grids automatically. Every k_l steps we regrid at level ℓ, which modifies grids at levels $\ell+1$ to ℓ_{max}. As in the initialization, error estimation and clustering are used to determine the new grids. Data for the new fine grids are copied from the previous grids at the same level if possible, otherwise the data are defined by interpolation from the underlying coarser grids.

Discretization of the projection

In this section we describe the numerical approximation to the projection. The projection is used to decompose a vector field V into an approximately divergence-free component V^d, and a component of the form $\frac{1}{\rho}\nabla\phi$. In the adaptive algorithm there are two distinct projections. The level projection operates on D_ℓ, the union of grids at level ℓ. The sync projection operates on two or more levels of the composite grid hierarchy. The basic approach in both cases is to solve for ϕ in

$$\nabla \cdot \frac{1}{\rho}\nabla\phi = \nabla \cdot V. \tag{4.1}$$

We will first describe the discrete approximations used for the elliptic operator and for $\nabla \cdot V$ in (4.1). The projection is derived from a finite element variational formulation. Specifically, we consider the scalar pressure field to be a C^0 function that is a bilinear function over each cell. For a detailed development of this approach the reader is referred to Almgren et al [1]. The discretization of (4.1) is obtained by solving the difference equations derived from the variational form

$$\int \frac{1}{\rho}\nabla\phi(\mathbf{x}) \cdot \nabla\psi_{i+\frac{1}{2},j+\frac{1}{2}}(\mathbf{x})\, d\mathbf{x} = \tag{4.2}$$

$$\int V \cdot \nabla\psi_{i+\frac{1}{2},j+\frac{1}{2}}(\mathbf{x})\, d\mathbf{x},$$

where the ψ's are the standard bilinear basis elements. The node values of ϕ obtained from solving (4.2) are used to update p and to determine the divergence-free component of V using

$$V^d = V - \frac{1}{\rho} G\phi, \tag{4.3}$$

where

$$\begin{aligned}(G\phi)_{ij} = \big(&\tfrac{1}{2\Delta x}\,(\phi_{i+1/2,j+1/2} + \phi_{i+1/2,j-1/2} \\ &\quad -\phi_{i-1/2,j+1/2} - \phi_{i-1/2,j-1/2}), \\ &\tfrac{1}{2\Delta y}(\phi_{i+1/2,j+1/2} + \phi_{i-1/2,j+1/2} \\ &\quad -\phi_{i+1/2,j-1/2} - \phi_{i-1/2,j-1/2})\,\big).\end{aligned}$$

The key issue in applying the weak formulation defined by (4.2) for both the level projection and the sync projection is the specification of what happens at internal boundaries between coarse and fine grids. Our approach has been to retain the notion that pressure is viewed as a continuous function that is bilinear on each cell, corresponding to a conforming finite element basis. At the coarse–fine boundary continuity requires that the fine grid values along the edge of a coarse cell be linearly interpolated from the coarse grid. Thus, the only unknowns at the coarse–fine boundaries are those corresponding to the nodes of coarse grid cells. The basis element ψ^{c-f} in (4.2) that is associated with a coarse–fine boundary node is therefore the piecewise bilinear function that is zero on all interior fine grid nodes and zero on all other coarse grid nodes but decreases linearly to zero along the fine grid nodes between the coarse node where it is one and adjacent coarse nodes where it is zero.

Before discussing the multigrid algorithm for solving (4.2), it will be helpful to reformulate the equation using the multilevel notation introduced in the previous section. For the level projection at level ℓ the values of ϕ on internal boundaries of the fine grid are given as Dirichlet conditions determined by $\phi^{\ell-1}$, the values of ϕ on the next coarser grid. In this case values of ϕ at level $\ell+1$ or higher do not play a role; the projection is performed as if higher level grids did not exist. For this case, we express (4.2) as

$$L^\ell(\phi^\ell, \phi^{\ell-1}) = D^\ell(V^\ell). \tag{4.4}$$

Here the operator L^ℓ is a density-weighted nine-point finite difference approximation to the continuous elliptic operator. The operator D^ℓ given by the right hand side of (4.2) defines a suitably scaled discrete divergence.

For the sync projection the values of ϕ at level ℓ depend not only on the values of $\phi^{\ell-1}$ for boundary values at internal boundaries of the level ℓ grids but also on values of $\phi^{\ell+1}$ on the interior of grids at level $\ell+1$. There is also an analogous dependence of the weak divergence on values of $V^{\ell+1}$. In both cases these dependencies arise because the basis element ψ^{c-f} defined at the coarse–fine interface has support in the fine grid. We express (4.2) in this case as

$$L^{\ell,\ell+1}(\phi^{\ell+1}, \phi^\ell, \phi^{\ell-1}) = D^{\ell,\ell+1}(V^\ell, V^{\ell+1}). \tag{4.5}$$

We note that this equation is defined only at locations in level ℓ grids that are not covered by interior nodes of level $\ell+1$ grids.

The linear system associated with the solution of (4.2) is the standard bilinear finite element stiffness matrix for a self-adjoint second-order elliptic operator. We solve this system using standard multigrid methods (see [14]) modified for use within the adaptive grid hierarchy. The multigrid algorithm for the level projection is straightforward since it only involves grids at a single level.

The sync projection involves multiple levels and so is more complex. The algorithm used here is similar to those developed by Almgren et al [2] and by Howell [13]. We will sketch the algorithm only for a two level sync projection for levels ℓ and $\ell+1$; extension to additional levels is straightforward. If the refinement ratio is greater than 2, for the multigrid solve we introduce additional temporary levels between levels ℓ and $\ell+1$. The basic multilevel relaxation cycle for the sync projection is:

1. Compute residual $D^{\ell+1} - L^{\ell+1}(\phi^{\ell+1}, \phi^\ell)$ on level $\ell+1$.

2. Relax on level $\ell+1$. If there are intermediate multigrid levels between levels ℓ and $\ell+1$, this relaxation takes the form of a "short" V-cycle, extending down to the coarsest intermediate level above level ℓ. Update $\phi^{\ell+1}$.

3. Evaluate composite grid residuals on both levels using (4.5) and restrict residuals on level $\ell+1$ to level ℓ.

4. Perform V-cycle on grids at level ℓ to obtain correction to ϕ^ℓ.

5. Interpolate correction to level $\ell+1$ and relax at level $\ell+1$ (again using a short V-cycle if necessary) to obtain correction to $\phi^{\ell+1}$.

This overall cycle is repeated until a desired level of reduction in the composite residual is obtained.

$16^2 - 32^2$	Rate	$32^2 - 64^2$	Rate	$64^2 - 128^2$
4.72e-3	2.38	9.26e-4	2.19	2.04e-4

Table 1: Convergence rates for velocity on refined grids. The top numbers refer to the grid size of the coarse grid in each calculation.

Typically, each cycle reduces the composite residual by almost an order of magnitude.

Numerical Results

In this section we present three sets of results. The first set demonstrates second-order convergence of the adaptive projection method. For the convergence study we use the same initial velocity profile as was used in [4]:

$$u(x,y) = -\sin^2(\pi x)\sin(2\pi y)$$

$$v(x,y) = \sin^2(\pi y)\sin(2\pi x)$$

in the unit square. The velocity satisfies homogeneous Dirichlet boundary conditions.

A single patch, centered in the domain and occupying one quarter of the area of the domain, is refined by a factor of two. We have turned off the error estimation routines and fixed the location of the patch for this study. Calculations were run to $t = 0.5$ with $\sigma = 0.5$ for a sequence of increasingly fine base grids. The difference in the solution between grids of adjacent resolution is proportional to the error on the coarser grid. The convergence rate is defined by finding the L_2 norm of the difference between the solution obtained on the refined patch for several pairs of base grids and then taking the $\log_2$ of the ratio of these norms. The convergence rates for velocity are given in Table 1.

The second computational example was designed to illustrate the robustness of the algorithm for handling the interface between coarse and fine grids. The initial conditions consist of a pair of counter-rotating vortices centered in a fine grid patch, which is refined by a factor of four over the base grid of 32x32. The regridding algorithm is turned off so the refined patch remains fixed. In Figure 1 we show the results as the pair crosses from the fine patch into the coarse domain. Note that no spurious vorticity is generated at the coarse-fine interface. In the last frame we see only the contours of the original vortices, now on the coarse grid. Although the vortices are somewhat smeared because the coarse grid is inadequate to resolve them, there is no vorticity created or left behind on the interface.

In Figure 2 we present results from the calculation of a helium-in-air bubble rise. This calculation was done using $r-z$ coordinates, with the bubble initially spherical and centered along the $r = 0$ axis (on the left in the figures shown). Gravity is the only external force. The base grid here is 16x32; there are two finer levels, each refined by a factor of four.

The initial conditions are zero velocity everywhere, and a density profile $\rho(r,z) = \alpha(1.225\text{x}10^{-3}) + (1-\alpha)(1.75\text{x}10^{-4})$ kg/cm^3, where $\alpha = \frac{1}{2}(1 + \tanh(\sqrt{(r^2+(z-z_0)^2)} - R_0)/W))$. In this problem, the physical domain is 4km x 8km, $z_0 = 1.4$km, $R_0 = 1$km, and $W = .04$km. Density contours are shown at times $t = 1.7$, 11.6, 22.9 and 41.4 seconds.

Note here that the hierarchical grid structure evolves with the flow, generating for this problem one grid at the middle level and anywhere from one to five grids at the finest level. Regridding occurred every two time steps.

The timings for the bubble rise problem indicate that the calculation took approximately 70 μ-seconds per cell on one processor of a Cray Y-MP 8/64. The finest calculation used in the convergence study averaged under 50 μ-seconds per cell.

Conclusions

We have developed a new adaptive projection method for time-dependent incompressible variable density flow. The levels in the adaptive mesh hierarchy are refined in both space and time. The advection step takes place on individual grids in an approach similar to that of the single grid method. The projection at each level is similar to the uniform grid projection, but must now incorporate multiple grids per level. In addition, we introduce a sync projection, which is needed to synchronize the solution at each level ℓ with the data at the levels above it at the end of each level ℓ time step. This adaptive projection method is second-order accurate and provides an accurate and efficient tool for modeling variable density flows. In future work, we plan to extend the method to three dimensions and include viscous transport.

References

[1] A. S. Almgren, J. B. Bell, and W. G. Szymczak. A numerical method for the incompressible Navier-Stokes equations based on an approximate projection. Technical Report UCRL-JC-112842, LLNL, January 1993.

[2] A. S. Almgren, T. Buttke, and P. Colella. A fast vortex method in three dimensions. In *10th AIAA Computational Fluid Dynamics Conference*, Honolulu, June 1991.

[3] J. B. Bell, M. J. Berger, J. S. Saltzman, and M. Welcome. Three dimensional adaptive mesh refinement for hyperbolic conservation laws. Technical Report UCRL-JC-108794, LLNL, Dec. 1991.

[4] J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for viscous, incompressible flow. In *8th AIAA Computational Fluid Dynamics Conference*, Honolulu, June 9–11, 1987.

[5] J. B. Bell, P. Colella, and L. H. Howell. An efficient second-order projection method for viscous incompressible flow. In *10th AIAA Computational Fluid Dynamics Conference*, Honolulu, June 24–27, 1991.

[6] J. B. Bell and D. L. Marcus. A second-order projection method for variable-density flows. *J. Comput. Phys.*, 101:334–348, 1992.

[7] J. B. Bell, J. M. Solomon, and W. G. Szymczak. A second-order projection method for the incompressible navier stokes equations on quadrilateral grids. In *9th AIAA Computational Fluids Dynamics Conference*, Buffalo, June 14–16, 1989.

[8] M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1989.

[9] M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.

[10] M. J. Berger and I. Rigoustsos. An algorithm for point clustering and grid generation. Technical Report NYU-501, New York University-CIMS, 1991.

[11] P. Colella. A direct Eulerian MUSCL scheme for gas dynamics. *SIAM Journal on Computing*, 6:104–117, January 1985.

[12] P. Colella. A multidimensional second order Godunov scheme for conservation laws. *J. Comput. Phys.*, 87:171–200, 1990.

[13] L. H. Howell. A multilevel adaptive projection method for unsteady incompressible flow. In *6th Copper Mountain Conference on Multigrid Methods*, Copper Mountain, CO, April 4–9 1993.

[14] P. Wesseling. *An Introduction to Multigrid Methods*. Wiley, New York, 1992.

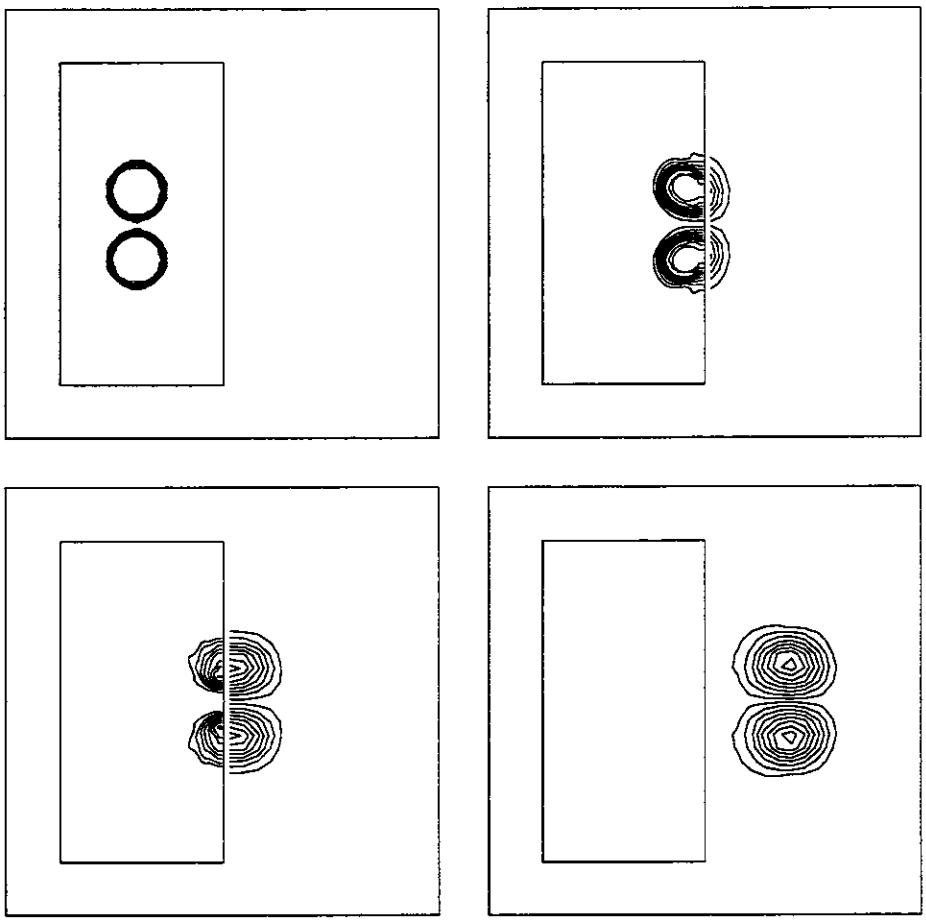

Figure 1: Counter-rotating vortex pair passing through a fixed coarse-fine grid interface with a refinement ratio of 4. Vorticity contours are shown after 0, 59, 76 and 121 coarse time steps.

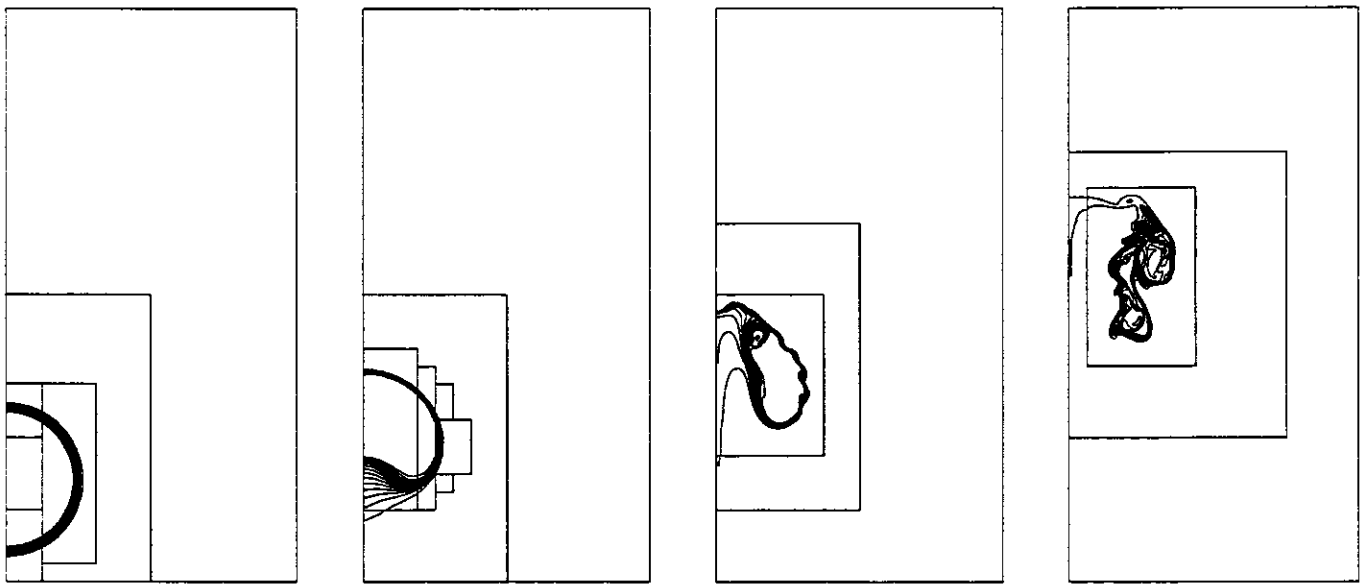

Figure 2: Helium bubble rising in air, modeled in r-z coordinates with two levels of mesh refinement—each by a factor of 4. Contours of density are shown.